基于生成对抗网络的恶意域名训练数据生成 *

袁　辰，钱丽萍，张　慧，张　婷

(北京建筑大学 电气与信息工程学院, 北京 100044)

摘　要： 当前僵尸网络大量采用DGA算法躲避检测，针对主流的基于人工规则的检测算法无法对最新产生的DGA域名进行识别检测和基于机器学习的检测算法缺乏演化的训练数据的问题，提出了一种基于Ascall编码方式定义域名编、解码器，并结合生成对抗网络构造域名字符生成器来预测生成DGA变体样本的方法。实验结果表明，在采用生成数据进行分类器训练和性能评估中，此方法生成的DGA域名变体样本可充当真实DGA样本，验证了生成数据的有效性并可用于DGA域名检测器的训练评估。

关键词： 恶意域名；DGA；生成对抗网络；检测；分类

中图分类号： TP393.08　　doi: 10.3969/j.issn.1001-3695.2017.12.0762

Generation of malicious domain training data based on generative adversarial network

Yuan Chen, Qian Liping, Zhang Hui, Zhang Ting

(*College of Electrical & Information Engineering*, *Beijing University of Civil Engineering & Architecture*, *Beijing* 100044, *China*)

Abstract: Many malware families such as botnet utilize domain generation algorithms(DGAs) to evade detection at present. The mainstream detection algorithms based on artificial rules and machine learning have some limitations due to the fact that DGAs generate domain characters timely and rapidly. The former is somewhat blind to new DGA variants. The latter suffers from the lack of evolving training data. In order to solve these problems, domain encoder and decoder on account of the method of Ascall encoding was defined in this paper and they were combined with the concept of generative adversarial network(GAN) to construct domain character generator. Then the generator was used to predict and generate DGA variants. Experiment results show that the DGA variants generated by this method can act as real DGA samples when these variants are utilized to train and estimate classifiers . This verifies the validity of the generated data and they can be effectively utilized to train and estimate DGA domain detector.

Key words: malicious domains; DGA; GAN; detection; classification

0　引言

随着互联网应用的快速发展，互联网承载的利益越来越大，各种网络攻击模式不断创新，网络安全事件的检测难度不断增大。木马和僵尸网络已成为变化形式最快、涉及范围最广、直接危害最重的网络威胁之一。据2016年CNCERT/CC抽样监测结果显示，2016年我国境内木马或僵尸程序控制服务器IP地址数量为48741个，较2015年上升了19.7%[1]。

域名系统（Domain Name System，DNS）作为互联网通信的基础服务，主要功能是将易于人为记忆理解的域名翻译成机器可以理解的主机IP地址。由于DNS服务的普遍性，攻击者大量注册恶意域名用于部署僵尸和木马程序，并为逃避基于恶意域名黑名单的检测，广泛采用域名生成算法技术（domain generation algorithm，DGA），亦称做域名变换技术（domain flux）来快速频繁变换域名[2]。DGA算法通过将操控僵尸网络主机的真实域名（又称C&C控制器）进行混淆和变换，掩饰真实主机的IP地址以躲避检测，大大降低了检测系统的检测能力。也因为恶意域名已成为网络僵尸和木马寄生的主要手段，对于恶意域名的识别检测一直是网络安全领域研究的重点和热点之一。

目前学术界针对DGA域名检测方法从多角度进行了探索，如统计学分析、主机行为分析、网络行为分析等。统计学分析主要考虑DGA域名的字符频率分布特性、主机域名访问数量等，主机行为分析主要考虑域名的客户端访问特性等，网络行为分析主要考虑网络流量或通信特征的变化或异常等。它们面

基金项目： 国家自然科学基金资助项目（61571144）；北京建筑大学博士基金资助项目（00331616014）

作者简介： 袁辰（1992-），男（通信作者），江苏徐州人，硕士研究生，主要研究方向为计算机网络安全（18210261356@163.com）；钱丽萍（1971-），女，安徽黄山人，副教授，博士，主要研究方向为网络安全、智能信息处理；张慧（1992-），女，河北秦皇岛人，硕士研究生，主要研究方向为网络安全；张婷（1994-），女，四川巴中人，硕士研究生，主要研究方向为网络安全.

临的共同局限是难以及时有效地获得足够的最新 DGA 域名训练样本数据，导致检测模型更新周期过长、过慢，检测的实效性、快速性不强。

本文在分析 DGA 域名的统计特性基础上结合深度神经网络[3]中的生成对抗网络（generative adversarial network，GAN[4]）对 DGA 域名进行生成预测分析，生成数据以扩大和预测训练样本，并通过实验验证了生成数据的有效性。

1　相关工作

从研究对象角度，目前 DGA 检测方法主要包括基于类目标检测和基于个目标检测两大类，前者主要基于域名的访问特性、访问时间等先对其进行聚类，再根据正常域名和 DGA 域名之间的字符统计分布差异，计算其间的距离（Jaccard 距离、编辑距离或 K-L 距离等）对聚类域名簇进行分类识别[5]。后者主要基于对单个域名的分析，如统计域名中元辅字符频度或长度、n-gram 正态分或频率等特性[5]。

从检测特征角度，主要有基于统计特征的方法、基于网络行为特征的方法、基于语法特征的方法及混合集成式方法等。Takeuchi Yuya 等人分析了 Water Tortue 的 DDoS 攻击，通过研究特定时间窗口内异常域名查询中快速变化域名块，利用域名块的 2-gram 分布、变化数量等，将相关特征输入贝叶斯分类器进行在线数据识别，准确率在 95.59%左右[6]。Tzy-Shiah Wang 等人文献基于被感染的僵尸主机会同时查询同一域中大量域名且仅有少数（C&C 控制主机）域名查询成功这一特性，针对基于 DGA 算法的 Botnet 难以检测和存活周期长等特点，提出了一种 DBod 的检测框架，主要基于 DNS 流量的查询行为进行分析[7]。Stefano Schiavoni 等人提出了 PHONENIX 探测机制，除了能够区分是否属 DGA 域名，还可进一步发现大量 DGA 域名背后隐藏的 Botnet，从而用以发现一些未知的 DGA 域名，作者采用大约 115 万域名进行验证，结果显示该机制的识别准确率在在 94.8%左右[8]。Yadav 等人从语言学的角度出发，针对 DGA 域名存在某些隐含的固定特性及正常域名内含的随机特性，提出了基于 DNS 流量的探测方法，主要从域名的字符数字分布和二元字符入手，对映射到同一 IP 地址的域名进行分类，通过计算域名之间的 K-L 距离、编辑距离和 Jaccard 距离等实现 DGA 域名的检测[9]。

从检测方法角度,已逐步从早期基于内容的 DPI、统计分析为主发展到以机器学习为主。SMARTbot 通过抽取与僵尸网络攻击相关的行为特征，对贝叶斯网络、SVM 等多种分类器横进行了评估[10]。DeepDGA[11]利用生成对抗网络对抗生成更难以检测的域名，以其逃避随机森林分类器检测。

综上所述，以上方法存在的主要不足在于难以实时检测最新生成的域名。同时，以数据驱动和基于机器学习检测方法日渐主流，用于生成检测模型的训练数据采集困难、环境普适性差、数据采集周期较长、模型更新演化迟滞，从而影响到检测器的在线检测性能。本文采用 GAN 神经网络，直接对 DGA 域名字符特征进行学习，无须预先对域名进行聚类、特征提取，只需对域名进行编码和解码，即可构造出和真实 DGA 样本域名相类似的生成域名。与文献[11]的不同之处在于：a）本文采用 DGA 域名训练 GAN 用于生成数据，训练和生成数据都更加具有针对性；b）本文为最大化利用 GAN 能直接对样本抽样学习的特性，不对数据做复杂的处理和变换（如不采用 CNN 层、pooling 层等），而是直接将数据输入 GAN 原始模型进行学习训练，以保持数据的真实特性；c）编解码器的构造具有简化和贴近原始数据的特性，从而最大化保持数据的真实特性；d）本文对生成域名样本采用更广泛的分类算法进行了分类验证，进一步验证了生成数据具备原始数据的特性和其有效性。

2　生成对抗网络

GAN 思想来源于博弈论中的纳什均衡[12]，其包含一对模型：生成模型（generative model，简称 G）和判别模型（discriminative model，简称 D）。

G 如同假币制造者，D 如同假币识别者，G 尽可能地学习真币的特征以提高自己欺骗 D 的手法，D 则尽可能地训练提升识别能力以避免被 G 欺骗。GAN 的学习过程就是 G 和 D 之间的一种竞争训练过程[4]。文献[4]将这一思想表示成式(1)：

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_{z(z)}}[\log(1-D(G(z)))] \tag{1}$$

式(1)也称为 min– max 公式，式中 $V(G,D)$ 为价值函数。对应的 GAN 神经网络模型如图 1 所示[13]。

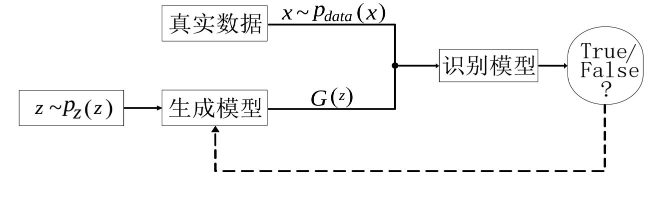


图 1　GAN 网络模型示意图

当将 GAN 训练用于数据生成时，假设存在真实数据 x（分类为 1）、生成数据 z （分类为 0），对于 D，最优的结果是将尽可能多的 x 判别为 1，将尽可能多的 z 判别为 0，即 $D(x)\approx 1$ 且 $D(G(z))\approx 0$，此时有 $\max_D V(G,D) =$ 。如果 x 被误判，即 $D(x)\approx 0$ 或 $D(G(z))\approx 1$，则有 $\log(D(x))\approx -\infty$ 或 $\log(1-D(G(z)))\approx -\infty$，此时 $V(G,D)-\infty$，所以 D 的学习过程就是不断提升 $V(D,G)$。对于 G，最优的结果是让 D 将尽可多的 X 判别为 0，将尽可能多的 Z 判别为 1，即 $D(x)\approx 0$ 且 $D(G(z))\approx 1$，此时有式（2）[4]。

$$\min_G V(G,D) = \max_G (E_{z\sim p_z}[\log(D(G(z)))]) \tag{2}$$

当 G 和 D 在训练中经多轮竞争最终达到平衡时

$D(G(z)) \approx 0.5$，此时真实数据和生成数据将非常相似。GAN 理论上可以完全逼近真实数据的分布模型，这是 GAN 神经网络的最大优势和特点。

3　DGA 域名字符生成模型

3.1　域名字符分析

理论上 GAN 中的生成器和判别器部分采用任意可微函数都能表示，因此其主要用于连续数据的处理，如图像生成、视频检测等[13]。基于文本的离散数据处理一直是深度神经网络研究的难点之一。本文基于字符串的文本域名来构建生成网络，在构造训练 GAN 之前，需要对域名数据样本做变换处理。

域名在构造上可分为两部分：主机名和域名（包括顶级域及可能的二级域、三级域等）。DGA 域名在构造上一般用随机算法来生成主机名，域名部分相对固定或变化较少。如 symmi 的 DGA 域名 hakueshoubar.ddns.net，其域名是由元辅音字符生成器和 ddns.net 组合而成；Conficker.C 的 DGA 域名 plrjgcjzf.net、gkrobqo.info 等也是由同频率的字符生成器和一级域名组合而成。因此本文中在生成域名时不考虑域名数据集中的一二级域名部分，只对 DGA 算法生成器的主机名的字符特性进行分析。

本文基于 GAN 的 DGA 域名数据生成模型主要包括域名编码器、生成网络、对抗网络和域名解码器四个部分。

3.2　域名编、解码器

假设去除顶级和二级域的域名字符为 d 顺序散列后组成的向量为 $\vec{d}$，即 $\vec{d} = d_1, d_2, \ldots d_i, \ldots, d_n]$，其中 n 为域名长度，$d_i (i = 1,2,\ldots,n)$ 为域名字符。字符 Ascall 码值转换函数为 $f(x) = A(x)$，域名字符向量 $\vec{d} = [d_1, d_2, \ldots d_i, \ldots, d_n]$ 可转换为形如 $A(\vec{d}) = [A(d_1), A(d_2), \ldots A(d_i), \ldots, A(d_n)]$ 的域名 Ascall 向量。为使 GAN 的学习效率更高，采用数据归一化将域名 Ascall 向量 $A(\vec{d})$ 的值映射到区间 $[0,1]$。对于 $i = 1,2,\ldots,n$，映射式如（3）所示：

$$A^*(d_i) = \frac{A(d_i) - \min A(d_i)}{\max A(d_i) - \min A(d_i)} \quad (3)$$

考虑到 ASCII 码表区间为 $[0,127]$，而区间 $[0,32]$ 中的字符值不能打印输出以及域名内无此种字符的特性，编码器映射函数的定义域取为 $[33,127]$，值域为 $[0,1]$，则 $\min A(d_i)$ 值为 33，$\max A(d_i)$ 值为 127。经上述映射后域名向量 $\vec{d}$ 被映射为 $\vec{d}^* = [A^*(d_1), A^*(d_2), \ldots A^*(d_i), \ldots, A^*(d_n)]$。例如域名 ampavhunh，域名字符向量 $\vec{d} = [a, m, p, a, v, h, u, n, h]$，则 $A(\vec{d}) = [97,109,112,97,118,104,117,110,104]$，编码后的域名向量 $\vec{d}^* = [0.673684,\ldots,0.673684,\ldots,0.747368]$。经此编码器编码后，字符域名向量转换为 GAN 的训练数据，最终通过 Tesorflow 转换为深度神经网络运算的张量。

域名张量还原成域名字符串。其实质是上述解码器的镜像。因此域名解码器的反向映射公式如式(4)所示。

$$A(d_i) = A(d_i) * [\max A(d_i) * \min A(d_i)] + \min A(d_i) \quad (4)$$

其中：$\max A(d_i)$ 为区间 $[33,127]$ 的上限，$\min A(d_i)$ 为区间的下限，$A^*(d_i)$ 为生成网络生成的域名字符向量中的元素。对于 ASCII 码值在区间 $[0,32]$ 内的元素，因其无法打印输出显示且域名中实际不含此类字符，故解码器对此类字符元素予以自动舍弃，只考虑区间 $[33,127]$ 内的字符元素。

假设由生成网络生成的域名向量 $\vec{d}^* = [A^*(d_1), A^*(d_2), \ldots A^*(d_i), \ldots, A^*(d_n)]$，解码后域名向量转换为 Ascall 向量 $A(\vec{d}) = [A(d_1), A(d_2), \ldots A(d_i), \ldots, A(d_n)]$，若假设 Ascall 码值函数的反函数为 $f(x) = A^{-1}(x)$，则经 $f(x)$,Ascall 向量 $A(\vec{d})$ 被映射为 $\vec{d} = [d_1, d_2, \ldots d_i, \ldots, d_n]$，将 $\vec{d}$ 中的元素 d_i 顺序组合后即为域名字符串 $d_1, d_2, \ldots d_i, \ldots, d_n$。

3.3　生成网络

生成网络由四层神经网络组成，包括输入层、隐含层和输出层，如图 4 所示。其中输入层数据来源于高斯分布模型并随机产生 n=100 维的数据，激活函数采用 ReLu 函数。网络包含两层隐含层，节点数分别为 n=150 和 n=300，激活函数亦采用 ReLu 函数。输出层节点数为 n=15（即域名向量维度），考虑到域名向量元素区间为 $[0,1]$，因此输出层的激活函数采用 sigmoid 函数。

3.4　判别网络

判别网络同样为四层神经网络，包括输入层、隐含层和输出层。其中输入层的数据来源有二，一部分来源于真实数据，另一部分来源于生成网络生成的生成数据，本文将域名长度设置为 15，因此输入数据维度 n=30。两层隐含层的节点数分别为 n=150 和 n=300，激活函数采用 ReLu 函数。输出层激活函数为 sigmoid 函数，数据在经过激活函数运算之前，将前 15 维数据和后 15 维数据拆分进行运算，分别输出真实数据和生成数据的 dropout[14]，即以一定概率随机丢弃，防止网络出现过拟合。

综上所述,由 编码器、解码器、生成网络、识别网络组成的网络模型如图 2 所示。

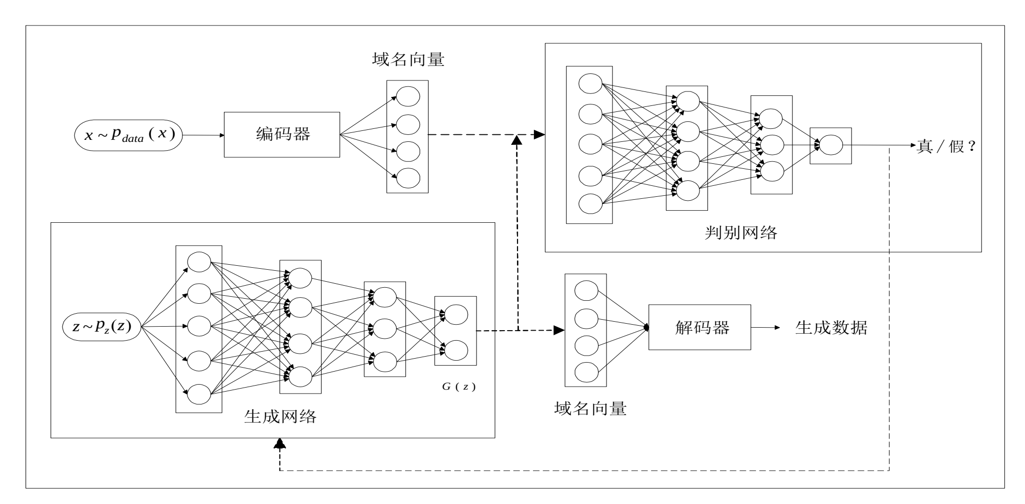


图 2　基于 GAN 的 DGA 域名字符生成网络模型

4　实验与分析

4.1　实验环境

本文中的实验环境主要包括实验平台和环境配置两部分。环境配置的详细信息如表 1 所示。

表 1　实验平台与环境配置

实验平台	环境配置
操作系统	Ubuntu 16.04
内存	4GB
CPU	Intel Corei5-3210 2.5GHz
编程语言	Python 2.7
深度学习框架	Tensorflow 0.12.0
机器学习平台	WEKA 3.8

4.2　数据集

数据集有四部分：100 万条 Conficker.C 真实 DGA 恶意域名样本、Alexa 排名前 5000 的负样本和真实 DGA 随机选取的 5000 个正样本、Alexa 排名前 5000 的负样本和生成类似 DGA 的 5000 个正样本、Alexa 排名前 10000 的负样本和 5000 个随机真实 DGA 样本与 5000 个随机生成的类似 DGA 样本组成的正样本。

选取划分以上数据集后，需要对其进行预处理。处理如下：

a）针对 DGA 域名的构成特性，采用 python 数组列表拆分函数 spit 对域名进行拆分，截取拆分后的前部分域名字符，去除顶级域及可能的二级域、三级域等，本部分处理主要包括百万级 DGA 恶意域名样本、Alexa 排名前 5000 的负样本和 DGA 随机选取的 5000 个正样本、Alexa 排名前 10000 的负样本，GAN 生成数据后续产生直接解码成字符，不需预处理。

b）百万级域名经过上述 a 处理后，为了缩短训练时间和减少 GAN 训练时的内存消耗，预先对域名字符进行数据编码与归一化处理，并通过 Tensorflow 中的数据标准读取格式转换成 GAN 神经网络的输入张量。

4.3　实验设计

本文在 GAN 模型的基础上尝试将 Ascall 编码方式与其相结合生成恶意域名训练数据，并通过分类器性能验证数据的有效性，实验设计如下：

a）类似 DGA 域名字符生成。本部分将预处理后的百万级域名输入域名字符生成模型，用于训练和生成类似 DGA 恶意域名样本。在每个网络训练的 epoch 内（1 个 epoch 等于使用训练集中的全部样本训练一次）生成网络产生出每次训练结束后的生成数据，每次产生 bach_size 个（批大小）列表数据。

b)特征选取。特征部分主要选用统计特征，包括域名长度、n-gram 频率（n=2、3、4、5）、n-gram 正态分[5]（n=2、3、4、5）、域名元音频率和域名辅音频率。

c）Alexa 负样本集与真实 DGA 正样本的分类。采用 b 中的特征对本数据集进行分类，此为后续两次分类结果的对比基准值，也是验证生成数据有效的基准。

d）Alexa 负样本集与生成类似 DGA 正样本集的分类。本部分分类同样采用 b 中的特征进行分类，分类结果与 c 中的分类结果进行比较，目的是为了验证类似样本可以充当 DGA 真实样本，从而说明生成数据的有效性。

e)Alexa 负样本集与真实 DGA 和生成类似 DGA 样本集的混淆分类。本部分分类同样采用 b 中的特征进行分类，将分类结果与 c 中的结果进行对比,此分类是为了说明在真实 DGA 与生成类似 DGA 混淆情况下，如果分类器结果良好，那么生成 DGA 数据具备真实 DGA 数据的特征，也验证了生成数据的有效性。

4.4　实验结果

4.4.1 类似 DGA 域名字符生成结果

为体现生成网络的学习特性，本文对不同学习阶段的生成数据进行了跟踪输出，生成数据的结果如图 3 所示。第一椭圆内的数据为真实 DGA 样本，第二椭圆内的数据为 GAN 对抗回合 0～10 产生的样本，第三椭圆内的数据为 GAN 对抗回合 250～253 产生的样本。

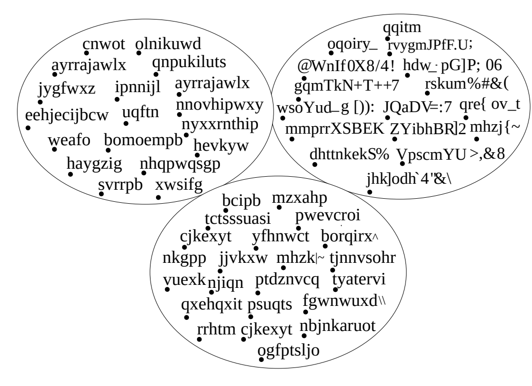


图 3　真实样本和不同对抗回合生成样本

4.4.2 分类验证结果

分类器选取 Weka 3.8 中的朴素贝叶斯、J48、随机树及随机森林，性能评估指标有正确率、错误率、精确率、F-measure 值及 ROC 面积。对三部分数据集的分类结果如表 2、4、6 所示，实际样本分类结果及分类模型的构建时间如表 3、5、7 所示。

表 2　Alexa 样本和真实 DGA 样本分类结果

分类器	正确率	错误率	精确率	F-Measure	ROC 面积
贝叶斯	0.999	0.001	0.999	0.999	1.00
J48	0.992	0.008	0.992	0.992	0.994
随机树	0.996	0.004	0.996	0.996	0.996
随机森林	0.997	0.03	0.997	0.997	1.00

表 3　样本分类的结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间
	正样本	负样本	正样本	负样本	
贝叶斯	4999	4992	1	8	0.55s
J48	4981	4940	19	60	0.57s
随机树	4986	4978	14	22	0.05s
随机森林	4995	4979	5	21	1.97s

表 4　Alexa 样本和类似 DGA 样本分类结果

分类器	正确率	错误率	精确率	F-Measure	ROC 面积
贝叶斯	0.984	0.016	0.984	0.984	0.998
J48	0.981	0.0.19	0.981	0.981	0.988
随机树	0.972	0.028	0.972	0.972	0.972
随机森林	0.983	0.017	0.983	0.983	0.999

表 5　样本分类的结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间
	正样本	负样本	正样本	负样本	
贝叶斯	4885	4983	145	17	0.1s
J48	4901	4907	99	93	0.19s
随机树	4860	4863	140	137	0.04s
随机森林	4926	4902	74	98	2.4s

表 6　Alexa 样本和混淆样本分类结果

分类器	正确率	错误率	精确率	F-Measure	ROC 面积
贝叶斯	0.988	0.012	0.989	0.988	0.999
J48	0.962	0.038	0.963	0.962	0.982
随机树	0.981	0.019	0.981	0.980	0.981
随机森林	0.989	0.011	0.989	0.989	1.00

表 7　样本分类的结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间
	正样本	负样本	正样本	负样本	
贝叶斯	9771	9995	229	5	0.29s
J48	9756	9491	244	509	0.57s
随机树	9818	9792	182	208	0.1s
随机森林	9928	9859	72	141	5.28s

4.5　实验结果分析

4.5.1 类似 DGA 域名字符生成结果分析

从能否作为域名的角度～来说，首先图 3 中真实数据是取自 Conficker.C 版本的恶意 DGA 域名预处理后的字符，为 GAN 需要学习的真实世界的数据；对抗回合 0～10 部分的数据是 GAN 在开始的对抗训练时生成的数据，此时产生的数据和真实数据差别很大，大部分数据不能作为域名的字符。GAN 在学习大约 250～253 对抗回合时，生成数据和真实数据开始趋于相似，生成数据中的大部分数据已经可以作为域名。

从字符分布的角度来说，对生成数据进行简单筛选与整理，剔除其中少部分不能作为域名的数据并进行一元字符统计分析，如图 4 所示。白色代表真实 DGA 样本字符频率分布，黑色代表 GAN 字符模型生成的样本频率。黑色数据围绕真实 DGA 样本上下波动，在经过 GAN 对抗训练后，频率分布在大样本下生成的 DGA 样本的频率围绕真实 DGA 的平均频率 0.0385 上下波动，因此，从字符分布特性的角度说明了类似 DGA 样本和真实 DGA 样本已经具有一定的相似性。



图 4　真实样本与生成样本一元频率分布

4.5.2 分类结果分析

由表 2、3 中的 Alexa 和真实 DGA 分类结果可以看出，本文采用的特征针对 Alexa 与真实 DGA 的样本分类时，朴素贝叶斯与随机森林分类效果较其他三种分类器良好。因此，首先说明选用以上描述的特征对于正负样本分类有效，其次，J48 和

随机树性能低于朴素贝叶斯和随机森林，但随机的森林的模型构建训练时间相对于其他分类器较长，时间复杂度较高。本文假设采用表 II 中真实数据样本的分类结果作为与 Alexa 样本和类似 DGA 分类、Alexa 样本和真实与类似 DGA 混淆样本分类的对比基准值。

由表 4、5 中 Alexa 和类似 DGA 分类结果可知，分类指标如正确率、错误率、召回率、精确率、F-measure 值、ROC 面积均与基准值保持在同一性能状态，说明在分类特征相同的情况下，生成的类似 DGA 样本可以充当真实 DGA 数据样本，从而说明了生成数据的有效性。

由表 6、7 中 Alexa 样本和混淆样本分类结果可知，在 Alexa 正常域名样本和分类特征不变的情况下，真实 DGA 样本和类似样本混淆分类器的指标如正确率、错误率、召回率、精确率、F-measure 值、ROC 面积仍与基准值处在同样的性能状态，说明类似样本已具备真实 DGA 样本的部分特性，也同样说明生成的类似样本有效。

综上所述，本部分从能否作为域名、域名的字符频率及多分类器效果对比三个层面说明通过 GAN 生成的数据既可以作为域名又具备 DGA 域名的特性，从而说明了数据的有效性。

5　结束语

恶意域名识别的数据集采集是网络安全领域中恶意域名的检测是中的重要任务之一，本文尝试将图像处理领域中的 GAN 对抗生成网络应用到网络安全中去生成恶意 DGA 域名字符数据集。解决恶意 DGA 域名的训练数据生成和识别检测问题，并通过实验初步验证了此方法的可行性。本文中为最大化利用 GAN 神经网络不用公式化描述数据分布和能够对原始数据直接进行学习的特性，本文将 DGA 域名字符进行简单的 Ascall 编码与数据归一化处理。其次，为了限制 GAN 网络生成数据过于自由化，本文编码器和解码器部分均对映射函数的定义域、值域部分进行限制，并对解码数据进行自动丢弃，从而让生成数据更符合真实样本数据。本文下一步工作将进一步研究如何改进编解码器以充分关联域名之间的字符序列特性，并评估其对 GAN 生成数据的质量影响和性能开销。

参考文献：

[1] 国家互联网应急处理协调中心. 2016 年中国互联网网络安全报告 [EB/OL]. (2017-5-27) [2017-12-15]. http://www. cert. org. cn/publish/main/46/index. html.

[2] 江健，诸葛建伟，段海新，等. 僵尸网络机理与防御技术. [J]. 软件学报，2012, 23 (1): 82-96.

[3] Goodfellow I J, Bengio Y, Courville A. Deep learning [J]. Genetic Programming & Evolvable Machines, 2017: 1-3.

[4] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [C]// Proc of International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2014: 2672-2680.

[5] Schiavoni S, Maggi F, Cavallaro L, et al. Phoenix: DGA-based otnet tracking and intelligence [C]// Proc of International Conference on etection of Intrusions and Malware, and Vulnerability ssessment. Springer, 2014: 192-211.

[6] Secure64. WaterTorture: a slow drip DNS DDoS attack [EB/OL]. (2015-11-30) [2017-12-15]. https://blog. secure64. com/?p=377.

[7] Wang T S, Lin H T, Cheng W T, et al. DBod: clustering and detecting DGA-based botnets using DNS traffic analysis [J]. Computers & Security, 2017, 64: 1-15.

[8] Schiavoni S, Maggi F, Cavallaro L, et al. Detection of intrusions and malware & vulnerability assessment [M]. Berlin: Springer, 2014: 192-211.

[9] Yadav S, Reddy A K K, Reddy A L N, et al. Detecting algorithmically generated malicious domain names [C]// Proc of ACM SIGCOMM Conference on Internet Measurement. New York: ACM Press, 2010: 48-61.

[10] Ahmad K, Roli S, Khurram K M. SMARTbot: a behavioral analysis framework augmented with machine learning to identify mobile botnet applications [J]. PLOS One, 2016, 11 (3): e0150077.

[11] Anderson H S, Woodbridge J, Filar B. DeepDGA: adversarially-tuned domain generation and detection [C]// Proc of ACM Workshop on Artificial Intelligence and Security. New York: ACM Press, 2016: 13-21.

[12] Ratliff L J, Burden S A, Sastry S S. Characterization and computation of local Nash equilibria in continuous games [C]// Communication, Control, and Computing. Piscataway: IEEE Press, 2013.

[13] 王坤峰，苟超，段艳杰，等. 生成式对抗网络 GAN 的研究进展与展望 [J]. 自动化学报，2017, 43 (3): 321-332.

[14] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15 (1): 1929-1958.